

# JSON WITH PERL

[http://www.tutorialspoint.com/json/json\\_perl\\_example.htm](http://www.tutorialspoint.com/json/json_perl_example.htm)

Copyright © tutorialspoint.com

This tutorial will teach you how to encode and decode JSON objects using Perl programming language. Let's start with preparing environment to start our programming with Perl for JSON.

## Environment

Before you start with encoding and decoding JSON using Perl, you will need to install JSON module which can be obtained from CPAN. Once you downloaded JSON-2.53.tar.gz or any other latest version, follow the following steps:

```
$tar xvfz JSON-2.53.tar.gz
$cd JSON-2.53
$perl Makefile.PL
$make
$make install
```

## JSON Functions

Function	Libraries
encode_json	Converts the given Perl data structure to a UTF-8 encoded, binary string.
decode_json	Decodes a JSON string.
to_json	Converts the given Perl data structure to a json string.
from_json	Expects a json string and tries to parse it, returning the resulting reference.
convert_blessed	Use this function with true value so that Perl can use TO_JSON method on the object's class to convert an object into JSON.

## Encoding JSON in Perl (encode\_json)

Perl encode\_json() function converts the given Perl data structure to a UTF-8 encoded, binary string.

### Syntax:

```
$json_text = encode_json ($perl_scalar );
or
$json_text = JSON->new->utf8->encode($perl_scalar);
```

## Example

The following example shows arrays under JSON with Perl:

```
#!/usr/bin/perl
use JSON;

my %rec_hash = ('a' => 1, 'b' => 2, 'c' => 3, 'd' => 4, 'e' => 5);
my $json = encode_json \%rec_hash;
print "$json\n";
```

While executing, this will produce following result:

```
{"e":5,"c":3,"a":1,"b":2,"d":4}
```

The following example shows how Perl objects can be converted into JSON:

```
#!/usr/bin/perl

package Emp;
sub new
{
    my $class = shift;
    my $self = {
        name => shift,
        hobbies => shift,
        birthdate => shift,
    };
    bless $self, $class;
    return $self;
}
sub TO_JSON { return { %{ shift() } }; }

package main;
use JSON;

my $JSON = JSON->new->utf8;
$JSON->convert_blessed(1);

$e = new Emp( "sachin", "sports", "8/5/1974 12:20:03 pm");
$json = $JSON->encode($e);
print "$json\n";
```

While executing, this will produce following result:

```
{"birthdate":"8/5/1974 12:20:03 pm","name":"sachin","hobbies":"sports"}
```

## Decoding JSON in Perl (decode\_json)

Perl decode\_json() function is used for decoding JSON in Perl. This function returns the value decoded from json to appropriate Perl type.

### Syntax:

```
$perl_scalar = decode_json $json_text

or

$perl_scalar = JSON->new->utf8->decode($json_text)
```

## Example

The following example shows how Perl can be used to decode JSON objects. Here you will need to install Data::Dumper module if you already do not have it on your machine.

```
#!/usr/bin/perl
use JSON;
use Data::Dumper;

$json = '{"a":1,"b":2,"c":3,"d":4,"e":5}';

$text = decode_json($json);
print Dumper($text);
```

While executing, this will produce following result:

```
$VAR1 = {  
    'e' => 5,  
    'c' => 3,  
    'a' => 1,  
    'b' => 2,  
    'd' => 4  
};
```